IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Krishna Kishore Yellepeddy, Lok Yan Leung, Anthony Joseph Nadalin		
Assignee:	International Business Machines Corporation		
Title:	Dynamic PKI Architecture		
Serial No.:	09/738,247	Filing Date:	December 15, 2000
Examiner:	Carl G. Colin	Group Art Unit:	2136
Docket No.:	AUS920000947US1	Customer No.	65362

Austin, Texas
June 12, 2007

COMMISSIONER FOR PATENTS
PO BOX 1450
ALEXANDRIA, VA 22313-1450

REQUEST FOR RECONSIDERATION AND RENEWED PETITION UNDER 37 CFR § 1.137(b)

Dear Sir:

In response to the Notice mailed May 1, 2007, dismissing Applicant's Petition filed on January 8, 2007, this is a Request for Reconsideration and Renewed Petition Under 37 CFR 1.137(b). This paper is also responsive to the Notice of Non-Compliant Amendment and Advisory Action Before the Filing of an Appeal Brief. Further examination and reconsideration are respectfully requested in view of the submission set forth below.

A. The Decision on the Petition to Revive Should Not Have Applied the Standard for Final Office Actions Since The April 21, 2005 Office Action Should Not Have Been Made "Final"

Though not entirely clear, the decision to dismiss the petition to revive appears to be based on the understanding that Applicants were responding to a final Office Action in the course of reviving the application. Thus, the dismissal decision was based on the determination that Applicant failed to submit a reply that *prima facie* placed the application in condition for allowance, as required by 37 CFR § 1.133 for responses to final office actions. Applicants respectfully submit that this understanding is not correct because the April 21, 2005 Office Action improperly entered a final rejection over the newly cited Schell, Balfanz and French references. A short review of the facts here confirms that the April 21, 2005 Office Action was

improperly designated as a final rejection because Applicants did not change the scope of invention in response to the first Office Action, so the newly cited references should not have been used to finally reject the claims. The relevant facts are as follows:

- In the first Office Action, the Examiner relied only on U.S. Patent No. 6,598,167 to Devine et al. to reject the claims as anticipated. *See*, Exhibit A (First Office Action (July 1, 2004)).
- In response to the first Office Action, Applicants made slight amendments to claims 1 and 13 to address an antecedent basis concerns noted by the Examiner. Other than typographical corrections, no other amendments were made to claims 14-50 (which included independent claim groups 24-34, 35-43 and 44-50. *See*, Exhibit B (Response to Office Action Under 37 C.F.R. § 1.111 (Jan. 3, 2005)).
- Stated once again, **no substantive amendments were made to claims 14-50 in response to the first Office Action.**
- In the second Office Action dated April 21, 2005, the Examiner relied on the new Schell, Balfanz and French references to provide "new ground(s) of rejection," but asserted (improperly) that "Applicant has changed the scope of the invention in view of the amended claims, and a new ground of rejection has been made in view of new references as discussed below." *See*, Exhibit C ("Final" Office Action, p. 2. (April 21, 2005)).

As seen from the foregoing, the Examiner improperly entered a final rejection in the April 21 Office Action since the Applicant responded to the first Office Action by pointing to deficiencies in the rejection analysis, and included only minor changes to **some** of the claims in order to correct certain claim informalities noted by the Examiner. Indeed, most of the claims were **not amended at all**, so it was improper for the Examiner to finally reject the claims based on the assertion that "Applicant has changed the scope of the invention in view of the amended claims." Final Office Action, p. 2 (April 21, 2005). Because the "final" rejection was improperly entered, Applicants request that the Petition to Revive be reconsidered in view of the response requirements for a non-final Office Action, which Applicants have plainly satisfied here. In particular, Applicants' amendments and arguments in the January 8, 2007 submission are sufficient to have avoided abandonment of a non-final Office Action. *See*, MPEP § 711.03(c)(II)(A)(2)(a) ("The required reply to a non-final action in a nonprovisional application abandoned for failure to prosecute may be … (A) an argument or an amendment under 37 CFR 1.111."). Based on the foregoing, Applicants request that the petition decision be reconsidered in light of the standard for non-final Office Actions, and that the petition to revive be granted.

B. In the Alternative, Applicants Request Reconsideration In View of the Notice of Appeal Submitted Herewith

In the event the Petitions Examiner maintains that the April 21, 2005 Office Action is a final Office Action, Applicants request, in the alternative, reconsideration of the decision on the petition to revive based on the Notice of Appeal submitted herewith. According to the relevant MPEP provision, a Notice of Appeal and appeal fee may be submitted to meet the reply requirement for consideration of a petition to revive in a non-provisional application abandoned for failure to reply to a final action. *See*, MPEP § 711.03(c)(II)(A)(2)(b) ("A reply under 37 CFR 1.113 to a final action must include a request for continued examination (RCE) under 37 CFR 1.114 or cancellation of, or appeal from the rejection of, each claim so rejected."). Through this submission, Applicants are making a *bona fide* attempt to provide a complete reply to the last Office action, and request that the Petition to Revive be reconsidered in view of the Notice of Appeal submitted herewith.

C. Applicants Request Entry of the Amendments Submitted Herewith

Since it is not clear from the "Advisory Action Before the Filing of an Appeal Brief" if the previous amendments were entered, Applicants hereby request that the amendments and arguments submitted hereinbelow be entered for purposes of the appeal. These amendments repeat the amendments to the drawings and specification that were submitted in the January 8, 2007 submission to address the drawing objections raised by the Examiner in the April 21, 2005 Office Action. In response to the Notice of Non-Compliant Amendment, these amendments also include the inadvertently omitted claim 8 which was previously presented. In addition, these amendments correct certain typographical errors in the claims. As a result, Applicants submit that these amendments are submitted to comply with a requirement of form expressly set forth in the April 21, 2005 Office Action, or at a minimum present the claims in better form for consideration on appeal. Accordingly, the amendments are permitted under 37 CFR § 1.116(b)(1)(2).

In the "Advisory Action Before the Filing of an Appeal Brief," the Examiner indicated that the drawing amendments had been considered, but apparently was not satisfied with the amendment because, according to the Examiner, "the specification on page 17 lines 18 and 23, and 27 still recite the same reference 300 to designate both system and network." In response, Applicants have amended the specification to remove the reference to "network 300" and replace it with "network 310" in keeping with the Figure 3 disclosure.

AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [0065] beginning on page 17, line 23 of the original application, with the following amended paragraph:

[0065] Likewise, if the request from the network [[300]] 310 is in the form of a Public Key Cryptography Standard #10 (PKCS10) format, a PKCS10 server bean 312 fields such a request and formats this type of request for transmittal to the remainder of the system 300.

Please replace paragraph [0071] beginning on page 19, line 18 of the original application, with the following amended paragraph:

[0071] Returning now to our exemplary system, the request now reaches an X.509 generator bean [[330]] 380. This bean [[330]] 380 generates the digital certificate based upon the X.509 specification that defines digital certificates containing signed user public keys.

Please replace paragraph [0093] beginning on page 24, line 5 of the original application, with the following amended paragraph:

[0093] As a final step in this example, the request is selectively directed to a particular path or bean from a selection of beans. This takes place in an IfThenElse bean 446, 448.

AMENDMENTS TO THE DRAWINGS

Applicants have enclosed nine (9) Replacement Sheets of formal drawings for this application. No new matter has been added.

Figure 3 has been amended to include omitted reference numeral 300 and to replace incorrect numeral 322 with 312.

Figure 4 has been amended to include reference number 400.

AMENDMENTS TO THE CLAIMS

1. (Previously Presented) An apparatus for implementing a request regarding a digital certificate in a distributed processing system, the apparatus comprising:

a request implementation software that implements a response to the request regarding the digital certificate in response to a propagated event object;

at least one reception bean, communicatively coupled to the request implementation software and the distributed processing system, that generates an event object in response to receiving the request to generate a digital certificate from the distributed processing system; and

the request implementation software instantiated in a real time executable object-oriented language.

2. (Currently Amended) The apparatus of claim 1, the at least one reception bean comprising a plurality of reception beans, and each of the plurality of reception beans generating an event in response to requests of differing formats.

3. (Original) The apparatus of claim 1, the request implementation software comprising at least one bean.

4. (Original) The apparatus of claim 3, the at least one bean comprising a pipe bean.

5. (Original) The apparatus of claim 3, the at least one bean comprising a bean implementing a test on the request.

6. (Original) The apparatus of claim 3, the at least one bean comprising a bean that alters the request.

7. (Original) The apparatus of claim 3, the at least one bean comprising a bean that publishes information regarding the request.

8. (Previously Presented) The apparatus of claim 3, comprising at least one sink bean and at least one pipe bean.

9. (Original) The apparatus of claim 3, the at least one bean comprising a sink bean.

10. (Original) The apparatus of claim 3, the at least one bean comprising a client bean that propagates a request in a first format.

11. (Original) The apparatus of claim 10, the at least one bean comprising another client bean that propagates a request in a second format.

12. (Original) The apparatus of claim 3, the certificate generation software comprising a legacy software.

13. (Previously Presented) A method for implementing a request regarding a digital certificate in a distributed processing system, the method comprising:

receiving the request to generate the digital certificate from the distributed processing system in an at least one reception bean;

generating a reception event object in response to step of receiving; propagating the reception event object;

selectively implementing a response to the request regarding the digital certificate in response to a propagated event object in a request implementation software;

the request implementation software instantiated in a real time executable object-oriented language.

14. (Original) The method of claim 13, the step of receiving comprising: receiving requests in differing formats; and the step of generating further comprising generating reception events in response to each request received.

15. (Original) The method of claim 13, the request implementation software comprising at least one bean.

16. (Original) The method of claim 15, the at least one bean comprising a pipe bean.

17. (Original) The method of claim 15, the step of selectively implementing comprising testing a parameter of the request.

18. (Original) The method of claim 15, the step of selectively implementing comprising altering a parameter of the request.

19. (Original) The method of claim 15, the step of selectively implementing comprising publishing information regarding the request.

20. (Original) The method of claim 15, the at least one bean comprising a sink bean.

21. (Original) The method of claim 15, the step of selectively implementing comprising propagating a request in a first format.

22. (Original) The method of claim 21, the step of selectively implementing comprising propagating a request in a second format.

23. (Original) The method of claim 15, the certificate generation software comprising a legacy software.

24. (Previously Presented) A computer program product on a computer usable medium, the computer usable medium having a computer usable program embodied therein for implementing a request regarding a digital certificate on a distributed data processing system, the computer usable program including:

instructions for receiving the request to regarding the digital certificate from the distributed processing system, the instructions for receiving instantiated in an at least one reception bean;

instructions for generating a reception event object in response to the instructions for receiving; instructions for propagating the reception event object;

instructions for selectively implementing a response to the request regarding the digital certificate in response to a propagated event object, the instructions for selectively implementing instantiated in a request implementation software; and

the instructions for receiving the request instantiated in a real time executable object-oriented language.

25. (Currently Amended) The computer program product of claim 24, the at least one reception bean comprising a plurality of reception beans, and each of the plurality of reception beans generating an event in response to requests of differing formats.

26. (Original) The computer program product of claim 24, the request implementation software comprising at least one bean.

27. (Original) The computer program product of claim 24, the at least one bean comprising a pipe bean.

28. (Original) The computer program product of claim 24, the at least one bean comprising a bean implementing a test on the request.

29. (Original) The computer program product of claim 24, the at least one bean comprising a bean that alters the request.

30. (Original) The computer program product of claim 24, the at least one bean comprising a bean that publishes information regarding the request.

31. (Original) The computer program product of claim 24, the at least one bean comprising a sink bean.

32. (Original) The computer program product of claim 24, the at least one bean comprising a client bean that propagates a request in a first format.

33. (Previously Presented) The computer program product of claim 24, the at least one bean comprising another client bean that propagates a request in a second format.

34. (Original) The computer program product of claim 24, the certificate generation software comprising a legacy software.

35. (Original) An apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising:

a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans; and

at least one of the plurality of beans comprising a pipe bean that propagates an event to another of the plurality of beans.

36. (Original) The apparatus of claim 35 further comprising a sink bean, the sink bean responsive to propagated events and consuming such propagated events.

37. (Original) The apparatus of claim 35 wherein the pipe bean passes the event to the another bean unaltered.

38. (Original) The apparatus of claim 35 wherein the pipe bean passes the event to the another bean in an altered format.

39. (Original) The apparatus of claim 35 further comprising a server bean, the server bean responsive to requests from the distributed processing system and generating events.

40. (Original) The apparatus of claim 35 further comprising a client bean, the client bean responsive to events from the other beans and generating requests to the distributed processing system.

41. (Original) The apparatus of claim 35 further comprising a generation bean, the generation bean generating a digital certificate in response to an event.

42. (Original) The apparatus of claim 35 further comprising a publisher bean, the publisher bean publishing information in response to an event.

43. (Original) The apparatus of claim 35 further comprising a filter bean, the filter bean filtering events based upon a predetermined criteria.

44. (Original) An apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising:

a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans; and

the respective events generated by the plurality of beans subclassing from a base class event.

45. (Original) The apparatus of claim 44 wherein the beans and events are written in a cross platform language.

46. (Original) The apparatus of claim 44 wherein the cross platform language is JAVA.

47. (Original) The apparatus of claim 44 wherein at least one of the beans is a publisher bean.

48. (Original) The apparatus of claim 44 wherein at least one of the beans is generator bean.

49. (Original) The apparatus of claim 44 wherein at least one of the beans is a server bean.

50. (Original) The apparatus of claim 44 wherein at least one of the beans is a client bean.

CONCLUSION

In view of the foregoing, Applicants hereby renew their petition to revive and respectfully request reconsideration of the decision. In particular, Applicants request that the petition should be granted because the pending April 21, 2005 Office Action was improperly entered as a final rejection, and should instead have been designated a non-final Office Action. Because Applicants' petition submission included the required reply for a non-final Office Action, the petition to revive should be granted. *See*, MPEP § 711.03(c)(II)(A)(2)(a) ("The required reply to a non-final action in a nonprovisional application abandoned for failure to prosecute may be … (A) an argument or an amendment under 37 CFR 1.111."). In the alternative, Applicant request reconsideration in view of the Notice of Appeal submitted herewith. *See*, MPEP § 711.03(c)(II)(A)(2)(b) ("A reply under 37 CFR 1.113 to a final action must include a request for … appeal from the rejection of each claim so rejected."). Applicants request further that the amendments submitted herewith be entered pursuant to 37 CFR § 1.116(b). Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the Examiner is requested to telephone the undersigned at 512-338-9100.

ELECTRONICALLY FILED
June 12, 2007

Respectfully submitted,

/Michael Rocco Cannatti/

Michael Rocco Cannatti
Attorney for Applicants
Reg. No. 34,791

H

Exhibit A - First Office Action (July 1, 2004)

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/738,247	12/15/2000	Krishna Kishore Yellepeddy	AUS9-2000-0947 US1	2751

7590 07/01/2004

Law Office of Joseph R. Burwell
P.O. Box 28022
Austin,, TX 78755-8022

EXAMINER	
COLIN, CARL G	
ART UNIT	PAPER NUMBER
2136	

DATE MAILED: 07/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/738,247	YELLEPEDDY ET AL.	
	Examiner	Art Unit	
	Carl Colin	2136	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 December 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-50* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-50* is/are rejected.

7)☒ Claim(s) *8* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 December 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. Pursuant to USC 131, claims 1-50 are presented for examination.

Specification

2. The preliminary amendments to the specification are not consistent with the page numbers of the disclosure except for the first paragraph starting "page 12...".

2.1 The disclosure is objected to because of the following informalities: on page 8, line 25, reference number "10" should be --102--. On page 17, there is a lack of consistency with "system 300" line 22 and "network 300", line 23. Appropriate correction is required.

2.2 The disclosure is objected to because of the following informalities: the copending applications mentioned on page 1 have no serial numbers.

2.3 The abstract of the disclosure is objected to because of the sentence "A architecture ...is described. Correction is required. See MPEP § 608.01(b). Line 20 also recites "in a environment", which requires correction.

Applicant is reminded of the proper language and format for an abstract of the disclosure.

The abstract should be in narrative form and generally limited to a single paragraph on a separate sheet within the range of 50 to 150 words. It is important that the abstract not exceed 150 words in length since the space provided for the abstract on the computer tape used by the printer is limited. The form and legal phraseology often used in patent claims, such as "means"

and "said," should be avoided. The abstract should describe the disclosure sufficiently to assist readers in deciding whether there is a need for consulting the full patent text for details.

The language should be clear and concise and should not repeat information given in the title. It should avoid using phrases which can be implied, such as, "The disclosure concerns," "The disclosure defined by this invention," "The disclosure describes," etc.

2.4 The disclosure is objected to because it contains embedded hyperlinks and/or other form of browser-executable codes (see p.13, line 18). Applicant is required to delete the embedded hyperlinks and/or other form of browser-executable codes. See MPEP § 608.01.

Drawings

3. Figure 3 is objected to as failing to comply with 37 CFR 1.84(p)(5) because it does not include reference numbers (300), (312) and (330) in the description on p. 17, line 18 and 27; page 17, line 25; and page 19, line 19 respectively. Appropriate correction is required.

Figure 3 is objected to as failing to comply with 37 CFR 1.84(p)(5) because it includes the reference sign: 322 not mentioned in the description. Appropriate correction is required.

3.1 Figure 4 is objected to as failing to comply with 37 CFR 1.84(p)(5) because it does not include reference numbers (400) in the description on p. 22, line 12. Appropriate correction is required.

Figure 4 is also objected to as failing to comply with 37 CFR 1.84(p)(5) because it includes the reference sign: 446 not mentioned in the description. Appropriate correction is required.

Applicant is required to carefully review the application to correct such errors.

A proposed drawing correction, corrected drawings, or amendment to the specification to add the reference sign(s) in the description, are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

4. **Claim 8** is missing from the application or Applicant misnumbered the claims. Appropriate correction is required.

4.1 **Claim 13** is objected to because of the following informalities: "the" reception software should be --a-- reception software.

4.2 **Claim 33** is objected to because of the following informalities: "the computer program product of claim 10" should be corrected because there is no antecedent basis in claim 33 to make it dependent of claim 10.

4.3 **Claim 24** is objected to because of the following informalities: the phrase "having computer a usable program" should be corrected. Appropriate correction is required.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

5.1 **Claims 1-7 and 9-50** are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent 6,598,167 to **Devine et al.**.

5.2 **As per claims 1, 3, 15, and 26, Devine et al.** discloses an apparatus for implementing a request regarding a digital certificate in a distributed processing system, the apparatus comprising: a request implementation software that implements a response to the request regarding the digital certificate in response to a propagated event object, for example (see column 12, line 16 through column 13, line 25 see also column 15, line 25 through column 16, line 34); at least one reception bean, communicatively coupled to the request implementation software and the distributed processing system, that generates an event object in response to receiving the request to generate a digital certificate from the distributed processing system, for example (see column 12, line 16 through column 13, line 25 and column 7, line 20 through

column 8, line 16 and figure 17 with description in columns 15-16); and the reception software instantiated in a real time executable object-oriented language, for example (see column 15, line 5 through column 16, line 34).

As per claims 13 and 24, Devine et al. discloses a method for implementing a request regarding a digital certificate in a distributed processing system, the method comprising: receiving the request to generate the digital certificate from the distributed processing system in an at least one reception bean, for example (see column 12, line 16 through column 13, line 25; see also column 15, lines 25 through column 16, line 15); generating a reception event object in response to step of receiving, for example (see column 12, line 16 through column 13, line 25; see also column 15, lines 25 through column 16, line 15); propagating the reception event object, for example (see column 15, line 25 through column 16, line 34); selectively implementing a response to the request regarding the digital certificate in response to a propagated event object in a request implementation software, for example (see column 15, line 25 through column 16, line 34 and see also column 17, lines 26-60); the reception software instantiated in a real time executable object-oriented language, for example (see column 15, line 5 through column 16, line 34).

As per claims 4, 16, 27, and 35, Devine et al. discloses an apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising: a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans, for example (see column 24, line 26 through column 25, line 67 and

column 18, lines 3-36 et seq.); and at least one of the plurality of beans comprising a pipe bean that propagates an event to another of the plurality of beans, for example (see column 18, lines 3-36 et seq.).

As per claim 44, Devine et al. discloses an apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising: a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans, for example (see column 24, line 26 through column 25, line 67 and column 18, lines 3-36 et seq.); and the respective events generated by the plurality of beans subclassing from a base class event, for example (see column 24, line 26 through column 25, line 67 and column 18, lines 3-36 et seq.).

As per claims 2, 14, and 25, **Devine et al.** discloses the limitation of the at least one reception bean comprising a plurality reception beans, and each of the plurality of reception beans generating an event in response to requests of differing formats, for example (see column 24, line 26 through column 25, line 67).

As per claims 5, 17, and 28, **Devine et al.** discloses the limitation of the at least one bean comprising a bean implementing a test on the request, for example (see column 25, lines 15-50).

As per claims 6, 18, 29, 37-39, and 49, Devine et al. discloses the limitation of, the at least one bean comprising a bean that alters the request, for example (see column 10, lines 67 and column 18, line 12-37 and column 15, line 25 through column 16, line 34 and see also column 17, lines 26-60).

As per claims 7, 19, 30, 42, and 47, Devine et al. discloses the limitation of the at least one bean comprising a bean that publishes information regarding the request, for example (see column 17, lines 25-60).

As per claims 9, 20, 31, and 36, Devine et al. discloses the limitation of the at least one bean comprising a sink bean, the sink bean responsive to propagated events and consuming such propagated events, for example (see column 5, line 57 through column 6, line 12).

As per claims 10, 21, and 32, Devine et al. discloses the limitation of the at least one bean comprising a client bean that propagates a request in a first format, for example (see column 5, lines 55-67).

As per claims 11, 22, and 33, Devine et al. discloses the limitation of the at least one bean comprising another client bean that propagates a request in a second format another client bean that propagates a request in a second format, for example (see column 10, lines 5-23 and column 13, line 25 through column 14, line 40).

As per claims 12, 23, and 34, Devine et al. discloses the limitation of the certificate generation software comprising a legacy software, for example (see column 5, line 57 through column 6, line 12).

As per claims 40 and 50, Devine et al. discloses the limitation of further comprising a client bean, the client bean responsive to events from the other beans and generating requests to the distributed processing system, for example (see column 5, line 57 through column 6, line 32).

As per claims 41 and 48, Devine et al. discloses the limitation of further comprising a generation bean, the generation bean generating a digital certificate in response to an event, for example (see column 12, lines 28-65).

As per claim 43, Devine et al. discloses the limitation of further comprising a filter bean, the filter bean filtering events based upon a predetermined criteria, for example (see column 25, lines 10-67).

As per claims 45-46, Devine et al. discloses the limitation of wherein the beans and events are written in a cross platform language, the cross platform language is JAVA, for example (see column 2, line 65 through column 3, line 8).

Conclusion

6. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure as the art discloses an object oriented processing system for implementing a request regarding a digital certificate.

US Patents: 5,862,325 Reed et al.

6,334,189 Granger et al.

6.1 Any inquiry concerning this communication or earlier communications from the examiner should be directed to Carl Colin whose telephone number is 703-305-0355. The examiner can normally be reached on Monday through Thursday, 8:00-6:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on 703-305-9648. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

cc

Carl Colin

Patent Examiner

June 24, 2004

AYAZ SHEIKH
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

TFW 2136

Exhibit B - Response to Office Action Under 37 C.F.R. § 1.111 (Jan. 3, 2005)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: **Yellepeddy et al.**	§	Group Art Unit: **2136**
	§	
Serial No.: **09/738,247**	§	Examiner: **Colin, C.**
	§	
Filing Date: **12/15/2000**	§	Atty. Docket #: **AUS9-2000-0947-US1**
	§	

For: **Dynamic PKI Architecture**

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to:
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450
on January 3, 2005.

By: _____
Joseph R. Burwell, Reg. No 44,468

RESPONSE TO OFFICE ACTION UNDER 37 C.F.R. § 1.111

The following remarks are offered in response to the Office Action mailed 07/01/2004; a petition for an extension of time is included with this response.

No additional fees are believed to be necessary for this response; if, however, any fees are necessary, please charge Deposit Account No. 50-1888 of Joseph Burwell to cover the cost of the fees.

I. Amendments to the Claims

Please amend the claims as follows with the following version of the claims in accordance with revised 37 CFR § 1.121.

1. (Currently Amended) An apparatus for implementing a request regarding a digital certificate in a distributed processing system, the apparatus comprising:

a request implementation software that implements a response to the request regarding the digital certificate in response to a propagated event object;

at least one reception bean, communicatively coupled to the request implementation software and the distributed processing system, that generates an event object in response to receiving the request to generate a digital certificate from the distributed processing system; and

the request implementation ~~reception~~ software instantiated in a real time executable object-oriented language.

2. (Original) The apparatus of claim 1, the at least one reception bean comprising a plurality reception beans, and each of the plurality of reception beans generating an event in response to requests of differing formats.

3. (Original) The apparatus of claim 1, the request implementation software comprising at least one bean.

4. (Original) The apparatus of claim 3, the at least one bean comprising a pipe bean.

5. (Original) The apparatus of claim 3, the at least one bean comprising a bean implementing a test on the request.

6. (Original) The apparatus of claim 3, the at least one bean comprising a bean that alters the request.

7. (Original) The apparatus of claim 3, the at least one bean comprising a bean that publishes information regarding the request.

8. (New) The apparatus of claim 3, comprising at least one sink bean and at least one pipe bean.

9. (Original) The apparatus of claim 3, the at least one bean comprising a sink bean.

10. (Original) The apparatus of claim 3, the at least one bean comprising a client bean that propagates a request in a first format.

11. (Original) The apparatus of claim 10, the at least one bean comprising another client bean that propagates a request in a second format.

12. (Original) The apparatus of claim 3, the certificate generation software comprising a legacy software.

13. (Currently Amended) A method for implementing a request regarding a digital certificate in a distributed processing system, the method comprising:

receiving the request to generate the digital certificate from the distributed processing system in an at least one reception bean;

generating a reception event object in response to step of receiving;

propagating the reception event object;

selectively implementing a response to the request regarding the digital certificate in response to a propagated event object in a request implementation software;

the request implementation ~~reception~~ software instantiated in a real time executable object-oriented language.

14. (Original) The method of claim 13, the step of receiving comprising:

receiving requests in differing formats; and

the step of generating further comprising generating reception events in response to each request received.

15. (Original) The method of claim 13, the request implementation software comprising at least one bean.

16. (Original) The method of claim 15, the at least one bean comprising a pipe bean.

17. (Original) The method of claim 15, the step of selectively implementing comprising testing a parameter of the request.

18. (Original) The method of claim 15, the step of selectively implementing comprising altering a parameter of the request.

19. (Original) The method of claim 15, the step of selectively implementing comprising publishing information regarding the request.

20. (Original) The method of claim 15, the at least one bean comprising a sink bean.

21. (Original) The method of claim 15, the step of selectively implementing comprising propagating a request in a first format.

22. (Original) The method of claim 21, the step of selectively implementing comprising propagating a request in a second format.

23. (Original) The method of claim 15, the certificate generation software comprising a legacy software.

24. (Currently Amended) A computer program product on a computer usable medium, the computer usable medium having ~~computer~~ a computer usable program embodied therein for implementing a request regarding a digital certificate on a distributed data processing system, the computer usable program including:

instructions for receiving the request to regarding the digital certificate from the distributed processing system, the instructions for receiving instantiated in an at least one reception bean;

instructions for generating a reception event object in response to the instructions for receiving;

instructions for propagating the reception event object;

instructions for selectively implementing a response to the request regarding the digital certificate in response to a propagated event object, the instructions for selectively implementing instantiated in a request implementation software; and

the instructions for receiving the request instantiated in a real time executable object-oriented language.

25. (Original) The computer program product of claim 24, the at least one reception bean comprising a plurality reception beans, and each of the plurality of reception beans generating an event in response to requests of differing formats.

26. (Original) The computer program product of claim 24, the request implementation software comprising at least one bean.

27. (Original) The computer program product of claim 24, the at least one bean comprising a pipe bean.

28. (Original) The computer program product of claim 24, the at least one bean comprising a bean implementing a test on the request.

29. (Original) The computer program product of claim 24, the at least one bean comprising a bean that alters the request.

30. (Original) The computer program product of claim 24, the at least one bean comprising a bean that publishes information regarding the request.

31. (Original) The computer program product of claim 24, the at least one bean comprising a sink bean.

32. (Original) The computer program product of claim 24, the at least one bean comprising a client bean that propagates a request in a first format.

33. (Currently Amended) The computer program product of claim 24, ~~10,~~ the at least one bean comprising another client bean that propagates a request in a second format.

34. (Original) The computer program product of claim 24, the certificate generation software comprising a legacy software.

35. (Original) An apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising:

a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans; and

at least one of the plurality of beans comprising a pipe bean that propagates an event to another of the plurality of beans.

36. (Original) The apparatus of claim 35 further comprising a sink bean, the sink bean responsive to propagated events and consuming such propagated events.

37. (Original) The apparatus of claim 35 wherein the pipe bean passes the event to the another bean unaltered.

38. (Original) The apparatus of claim 35 wherein the pipe bean passes the event to the another bean in an altered format.

39. (Original) The apparatus of claim 35 further comprising a server bean, the server bean responsive to requests from the distributed processing system and generating events.

40. (Original) The apparatus of claim 35 further comprising a client bean, the client bean responsive to events from the other beans and generating requests to the distributed processing system.

41. (Original) The apparatus of claim 35 further comprising a generation bean, the generation bean generating a digital certificate in response to an event.

42. (Original) The apparatus of claim 35 further comprising a publisher bean, the publisher bean publishing information in response to an event.

43. (Original) The apparatus of claim 35 further comprising a filter bean, the filter bean filtering events based upon a predetermined criteria.

44. (Original) An apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising:

a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans; and

the respective events generated by the plurality of beans subclassing from a base class event.

45. (Original) The apparatus of claim 44 wherein the beans and events are written in a cross platform language.

46. (Original) The apparatus of claim 44 wherein the cross platform language is JAVA.

47. (Original) The apparatus of claim 44 wherein at least one of the beans is a publisher bean.

48. (Original) The apparatus of claim 44 wherein at least one of the beans is generator bean.

49. (Original) The apparatus of claim 44 wherein at least one of the beans is a server bean.

50. (Original) The apparatus of claim 44 wherein at least one of the beans is a client bean.

II. Amendment to the Abstract

Please amend the abstract by deleting or canceling the previous abstract and replacing it with the version of the abstract on the following page.

ABSTRACT OF THE DISCLOSURE

DYNAMIC PKI ARCHITECTURE

An architecture for implementing PKI technology is described. Individual software module building blocks, or "beans", are responsive to events and are placed and linked together in an assembly-line-like manner. Each bean is responsive to particular events and does one particular action in the scheme. For example, individual beans are responsive to different format PKI requests from a network, and, in turn, generate an event corresponding to that request. The event is broadcast to other beans that take the event and perform some other operation in the defined process. Other beans include certificate generators, publishers, manipulators, broadcasters to output streams, and also beans that can act as boolean branches. When strung together, the beans form a cohesive PKI schema. The beans may be used to build both defined and customized PKI schemas easily. As new certificate standards or new dissemination methods are developed, appropriate beans may be written to implement these.

III. Amendments to the Specification

Please amend the specification by deleting the noted original paragraphs and replacing them with the following versions of the paragraphs.

Page 1, lines 4-10:

The following co-pending applications are all assigned to the assignee of the invention and are incorporated herein by reference:

1. U.S. Serial No. 09/738,240, ~~_____~~, ~~entitled~~ "Configurable PKI Architecture" filed on 12/15/2000; ~~_____ in the names of _____;~~ ~~and~~

2. U.S. Serial No. 09/738,239, ~~_____~~, ~~entitled~~ "Dynamic Modular PKI Architecture" filed on 12/15/2000. ~~_____ in the names of _____;~~

Page 8, lines 24-30:

Additional computing components connected to the network 102 ~~10~~ may include a personal digital assistant 114 and a remote network appliance 116. Additionally, an individual user may carry a so-called "smart card" 118. The smart card may contain sufficient data and/or processing capabilities to allow connection to and communication with other components of the distributed data processing system 100.

Page 12, lines 1-10:

Typically, after the CA has received a request for a new digital certificate, which contains the requesting entity's public key, the CA puts ~~signs~~ the requesting entity's public key into a certificate and signs the certificate with the CA's private key. ~~with the CA's private key and places the signed public key within the digital certificate.~~ Anyone who receives the digital certificate during a transaction or communication can then use the public key of the CA to verify the signed public key within the certificate. The intention is that an entity's certificate verifies that the entity owns a particular public key.

Page 13, lines 11-18:

Other aspects of certificate processing are also standardized. The Certificate Request Message Format (RFC 2511) specifies a format recommended for use whenever a relying party is requesting a certificate from a CA. Certificate Management Protocols have also been promulgated for transferring certificates. More information about the X.509 public key infrastructure (PKIX) can be obtained from the Internet Engineering Task Force (IETF) at "www.ietf.org". ~~www.ietf.org.~~

Page 14, lines 1-18:

Figure 2 is a block diagram depicting a typical manner in which an individual obtains a digital certificate. User 202, operating on some type of client computer, has previously obtained or generated a public/private key pair, e.g., user public key 204 and user private key 206. User 202 generates a request for certificate 208 containing user public key 204 and sends the request to certifying authority 210, which is in possession of CA public key 212 and CA private key 214. Certifying authority 210 verifies the identity of user 202 in some manner and generates X.509 digital certificate 216 containing ~~signed~~ user public key 218, ~~216 that was~~ and the certificate is signed with CA private key 214. User 202 receives newly generated digital certificate 216, and user 202 may then publish digital certificate 216 as necessary to engage in trusted transactions or trusted communications. An entity that receives digital certificate 216 may verify the signature of the CA by using CA public key 212, which is published and available to the verifying entity.

Page 16, lines 1-9:

Furthermore, a specific PKI solution may be built from the ground up. For example, one application may need a simple CA that is able to issue certificates and manage the life cycle of the ~~these~~ certificates including revocation of certificates and generating certificate revocation lists. Another application used by the same entity may also issue certificates in bulk and have support for multiple applications and/or identification procedures. As such, a readily adaptable and lightweight system is described herein for the implementation of any such PKI technology.

Page 17, lines 23-27:

Likewise, if the request from the network 310 ~~300~~ is in the form of a Public Key Cryptography Standard #10 (PKCS10) format, a PKCS10 server bean 322 ~~312~~ fields such a request and formats this type of request for transmittal to the remainder of the system 300.

Page 18, lines 20-28:

After clearing the auditor bean 360, the request may then be sent to a Lightweight Directory Access Protocol (LDAP) publisher bean 370. This LDAP publisher bean 370 publishes certificates or certificate revocation lists, which may be obtained in ~~any specific parameters associated with such a request to an LDAP directory structure 373. Again, this may take place as the request winds its way through the PKI request system 300 an initial time. Or the publishing of the request in the LDAP directory 373 may take place in~~ the return path of the requests ~~request~~ after reaching the terminus bean.

Page 19, lines 18-22:

Returning now to our exemplary system, the request now reaches an X.509 generator bean 380. ~~330.~~ This bean 380 ~~330~~ generates the digital certificate based upon the X.509 specification that defines digital certificates containing signed user public keys.

Page 24, lines 5-8:

As a final step in this example, the request is selectively directed to a particular path or bean from a selection of beans. This takes place in an IfThenElse bean 446 or 448.

IV. General Remarks Concerning This Response

Claims 1-50 are currently pending in the present application. Claims 1, 13, 24, and 33 have been amended to fix antecedent basis errors or typographical errors. No claims have been canceled herein.

Claim 8, which was missing in the original set of claims, has been added herein.

Claims 1 and 13, which incorrectly mentioned "the reception software" in the original claims (which was noted as lacking an antecedent basis by the Office action), have been corrected to refer to the antecedent term "the request implementation software" in both claims.

Applicant notes that the claims have not been amended to avoid prior art; all of the amendments to the claims were minor corrections that did not substantively change the subject matter in the claims. Reconsideration of the claims is respectfully requested.

Several errors in the specification concerning reference numbers were noted by the Office action; the specification has been corrected herein.

Errors in Figure 3 and Figure 4 that were noted by the Office action have been corrected. A set of formal drawings are being submitted by mail separately from this response, which is being faxed.

The Office action also noted that the abstract was too long; a new abstract is being submitted herein.

V. 35 U.S.C. § 102(e)—Anticipation—*Devine et al.*

The Office action has rejected claims 1-7 and 9-50 under 35 U.S.C. § 102(e) as anticipated by Davis, "Secure Customer Interface for Web Based Data Management", U.S. Patent No. 6,598,167 B2, filed 09/26/1997, issued 07/22/2003. This rejection is respectfully traversed.

The claims of the present patent application are directed to a particular software architecture comprising beans that respond to propagated events for implementing functionality related to digital certificates; each of the independent claims introduces a requirement of at least one bean. However, the claim rejections completely ignore the specific language in the claims that recite these software architectural features. For example, the second element of amended independent claim 1 reads as follows:

> at least one reception bean, communicatively coupled to the request implementation software and the distributed processing system, that generates an event object in response to receiving the request to generate a digital certificate from the distributed processing system;

The rejection of independent claim 1 states that this feature is found in Devine et al. at column 12, line 16, to column 13, line 25, and at column 7, line 20, through column 8, line 16, along with Figure 17 and its description in columns 15-16.

Applicant strongly disagrees that Devine et al. discloses the claimed features. Moreover, if Devine et al. disclosed each of the claimed features as argued in the rejections, then the rejection should have been written so that there was a clear correspondence between disclosed features and claimed features. However, the rejection confusingly points to several columns of text for most features without presenting any additional arguments that explain how a specific feature of Devine et al., e.g., as might be found within a few lines of text in Devine et al., discloses a specific feature of the claimed invention. By pointing to several columns of text, the rejection obfuscates the fact that Devine et al. does not disclose the claimed features. Applicant asserts that the rejection purposefully points to large amounts of text for a given claimed feature of the present invention in order to appear as having a correct format for an anticipation rejection while completely lacking substantive merit.

In order to emphasize that Devine et al. does not disclose the claimed features, Applicant copies hereinbelow the portions of Devine et al. that have been applied by the rejection against the second element of claim 1 along with other portions of Devine et al. that have been applied against other claims.

Column 7, line 20, to column 8, line 16:

FIG. 3 illustrates an example client GUI presented to the client/customer as a browser web page 60 providing, for example, a suite 70 of network management applications, which may include: Traffic Monitor 72; an Alarm Monitor 73; a Network Manager 74 and Intelligent Routing 75. Access to network functionality is also provided through Report Requester 76, which provides the ability to define and request a variety of reports for the client/customer and a Message Center 77 for providing enhancements and functionality to traditional e-mail communications by providing access to user requested reports and bulk data. Additional network MCI Internet applications not illustrated in FIG. 3 include Online Invoice, relating to electronic invoicing and Service Inquiry related to Trouble Ticket Management.

As shown in FIGS. 2 and 3, the browser resident GUI of the present invention implements a single object, COBackPlane which keeps track of all the client applications, and which has capabilities to start, stop, and provide references to any one of the client applications.

The backplane 12 and the client applications use a browser 14 such as the Microsoft Explorer versions 4.0.1 or higher for an access and distribution mechanism. Although the backplane is initiated with a browser 14, the client applications are generally isolated from the browser in that they typically present their user interfaces in a separate frame, rather than sitting inside a Web page.

The backplane architecture is implemented with several primary classes. These classes include COBackPlane, COApp, COAppImpl, COParm. and COAppFrame classes. COBackPlane 12 is an application backplane which launches the applications 54a, 54b, typically implemented as COApp. COBackPlane 12 is generally implemented as a Java applet and is launched by the Web browser 14. This backplane applet is responsible for launching and closing the COApps.

When the backplane is implemented as an applet, it overrides standard Applet methods unit( ), start( ), stop( ) and run( ). In the unit( ) method, the backplane applet obtains a COUser user context object. The COUser object holds information such as user profile, applications and their entitlements. The user's configuration and application

entitlements provided in the COUser context are used to construct the application toolbar and Inbox applications. When an application toolbar icon is clicked, a particular COApp is launched by launchApp( ) method. The launched application then may use the backplane for inter-application communications, including retrieving Inbox data.

The COBackPlane 12 includes methods for providing a reference to a particular COApp, for interoperation. For example, the COBackPlane class provides a getApp( ) method which returns references to application objects by name. Once retrieved in this manner, the application object's public interface may be used directly.

The use of a set of common objects for implementing the various functions provided by the system of the present invention, and particularly the use of browser based objects to launch applications and pass data therebetween is more fully described in the above referenced copending application GRAPHICAL USER INTERFACE FOR WEB ENABLED APPLICATIONS, and Appendix A, attached to that application, provides descriptions for the common objects which includes various classes and interfaces with their properties and methods.

Column 12, line 16, through column 13, line 25:

Another communications issue involving the secure communications link, is the trust associated with allowing the download of the Java common objects used by the present invention, as discussed earlier with respect to the browser, since the Java objects used in the present invention require that the user authorize disk and I/O access by the Java object.

Digital Certificates, such as those developed by VeriSign, Inc. entitled Verisign Digital ID.TM. provide a means to simultaneously verify the server to the user, and to verify the source of the Java object to be downloaded as a trusted source as will hereinafter be described in greater detail.

As illustrated in FIG. 10, the process starts with the browser launch as indicated at step 280, and the entry of the enterprise URL, such as HTTPS://www.enterprise.com as indicated at step 282. Following a successful connection, the SSL handshake protocol is initiated as indicated at step 283. When a SSL client and server first start communicating, they agree on a protocol version, select cryptographic algorithms, authenticate the server (or optionally authenticate each other) and use public-key encryption techniques to generate shared secrets. These processes are performed in the handshake protocol, which can be summarized as follows: The client sends a client hello message to which

the server must respond with a server hello message, or else a fatal error will occur and the connection will fail. The client hello and server hello are used to establish security enhancement capabilities between client and server. The client hello and server hello establish the following attributes: Protocol Version, Session ID, Cipher Suite, and Compression Method. Additionally, two random values are generated and exchanged: ClientHello.random and ServerHello.random.

Following the hello messages, the server will send its digital certificate. Alternately, a server key exchange message may be sent, if it is required (e.g. if their server has no certificate, or if its certificate is for signing only). Once the server is authenticated, it may optionally request a certificate from the client, if that is appropriate to the cipher suite selected.

The server will then send the server hello done message, indicating that the hello-message phase of the handshake is complete. The server will then wait for a client response. If the server has sent a certificate request Message, the client must send either the certificate message or a no_certificate alert. The client key exchange message is now sent, and the content of that message will depend on the public key algorithm selected between the client hello and the server hello. If the client has sent a certificate with signing ability, a digitally-signed certificate verify message is sent to explicitly verify the certificate.

At this point, a change cipher spec message is sent by the client, and the client copies the pending Cipher Spec into the current Cipher Spec. The client then immediately sends the finished message under the new algorithms, keys, and secrets. In response, the server will send its own change cipher spec message, transfer the pending to the current Cipher Spec, and send its finished message under the new Cipher Spec. At this point, the handshake is complete and the client and server may begin to exchange user layer data.

Column 15, line 25, to column 16, line 34:

FIG. 7 is a diagram which illustrates a security module design having clean separation from the browser specific implementations. The security module includes the main COSecurity class 402, and the interface COBrowserSecurityInterface 404. The COSecurity object checks browser type upon instantiation. It does so by requesting the "java.vendor" system property. In the preferred embodiment of the invention, Microsoft Internet Explorer.TM. is the default browser, but if the browser is Netscape, for

example, the class then instantiates by name the concrete implementation of the Netscape security interface, nmco.security.securityimpls. CONetscape4.sub.-- OSecurityImpl 406. Otherwise, it instantiates nmco.security.securityimpls. CODefaultSecurityImpl 408.

The COBrowserSecurityInterface 404 mirrors the methods provided by COSecurity 402. Concrete implementations such as CONetscape4.sub.-- OSecurityImpl 406 for Netscape Communicator and CODefaultSecurityImpl 408 as a default are also provided. Adding a new implementation 410 is as easy as implementing the COBrowserSecurityInterface, and adding in a new hook in COSecurity.

After using "java.vendor" to discover what browser is being used, COSecurity 402 instantiates by name the appropriate concrete implementation. This is done by class loading first, then using Class.newInstance( ) to create a new instance. The newInstance( ) method returns a generic object; in order to use it, it must be cast to the appropriate class. COSecurity 402 casts the instantiated object to COBrowserSecurityInterface 404, rather than to the concrete implementation. COSecurity 402 then makes calls to the COBrowserSecurityInterface "object," which is actually a concrete implementation "in disguise." This is an example of the use of object oriented polymorphism. This design cleanly separates the specific implementations which are browser-specific from the browser-independent COSecurity object.

Each COApp object may either create their own COSecurity object using the public constructors, or retrieve the COSecurity object used by the backplane via COBackPlane.getSecurity( ). In general, the developer of the applications to be run will use the COSecurity object whenever the COApp needs privileged access to any local resource, i.e., access to the local disk, printing, local system properties, and starting external processes. The following represents an example of the code generated when using the security object.

```
// Instantiating CoSecurity objectCOSecurity
security=new CoSecurity( );
// Now access a privileged resource
try {
     String s=
     security.getSystemProperty("user.home");
     System.out.println(s);
}
catch(COSecurityException cose)
{
// take care in case of security exception
}
```

> Referring back to FIG. 10, once the browser type has been confirmed, the logon applet checks for the name/password entry and instantiates a session object in step 292, communicating the name/password pair to the enterprise system. The session object sends a message containing the name/password to the StarOE server 49 for user validation in step 294.
>
> When the user is properly authenticated by the server in step 296, another Web page which launches the backplane object is downloaded in steps 298, 300, 304. This page is referred to as a home page. At the same time, all the remaining application software objects are downloaded in CAB or JAR files as indicated at step 302. If the system of the present invention determines that the backplane and application files have been already downloaded, the steps 300, 302, 304 are not performed. The backplane object is then instantiated in step 306.

As should be apparent by merely reading the above-copied portions of Devine et al., the portions of Devine et al. that have been applied against the second element of independent claim 1 do not disclose the use of "a reception bean", as required by the claim language. In fact, Devine et al. does not even mention the use of a bean at all. At most, Devine et al. discloses object-oriented classes and objects; however, a bean is not identical nor equivalent with a generic object-oriented class or object because a bean is an object that has particular operational properties or characteristics. Since Devine et al. does not disclose the use of beans, Devine et al. cannot be used as an anticipatory reference against the claims. At a minimum, since Devine et al. cannot be used as an anticipatory reference, the Office action should have at least attempted to explain how the teachings of Devine et al. could have hypothetically been modified in an obvious manner to reach the claimed invention, which would require an obviousness rejection rather than an anticipatory rejection.

A common rejection was applied against independent claim 1 and dependent claims 3, 15, and 26. Claims 3, 15, and 26 recite that the request implementation software comprises at least one bean. Since Devine et al. does not disclose the use of beans, Devine et al. cannot be used as an anticipatory reference against claims 3, 15, and 26.

Whereas independent claim 1 is directed to an apparatus, independent claim 13 is directed to a method, and independent claim 24 is directed to a computer program product. A common rejection was applied against claims 13 and 24. Claims 13 and 24 contain additional elements, but the rejection of claims 13 and 24 relies on the same argument as the rejection of claim 1 and points to the same portions of Devine et al.. Applicant's argument with respect to the rejection of claim 1 is applicable against claims 13 and 24.

Independent claim 35 reads as follows:

> An apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising:
> a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans; and
> at least one of the plurality of beans comprising a pipe bean that propagates an event to another of the plurality of beans.

In order to emphasize that Devine et al. does not disclose the claimed features, Applicant copies hereinbelow the portions of Devine et al. that have been applied by the rejection against claim 35.

Column 18, line 3, to column 19, line 22:

> FIG. 11 is a data flow diagram illustrating data flow among the processing modules of the "network MCI Interact" during logon, entitlement request/response, heartbeat transmissions and logoff procedures. As shown in FIG. 11, the client platform includes the networkMCI Interact user 340 representing a customer, a logon Web page having a logon object for logon processing 342, a home page having the backplane object. The Web server 344, the dispatcher server

346, cookie jar server 352, and StarOE server 348 are typically located at the enterprise site.

As described above, following the SSL handshake, certain cab files, class files and disclaimer requests are downloaded with the logon Web page as shown at 440. At the logon Web page, the customer 340 then enters a userid and password for user authentication as illustrated at 440. The customer also enters disclaimer acknowledgment 440 on the logon page 342. If the entered userid and password are not valid or if there were too many unsuccessful logon transactions, the logon object 342 communicates the appropriate message to the customer 340 as shown at 440. A logon object 342, typically an applet launched in the logon Web page connects to the Web server 344, for communicating a logon request to the system as shown at 442. The logon data, having an encrypted userid and password, is sent to the dispatcher 346 when the connection is established as shown at 444. The dispatcher 346 then decrypts the logon data and sends the data to the StarOE 348 after establishing a connection as shown at 446. The StarOE 348 validates the userid and password and sends the results back to the dispatcher 346 as illustrated at 446 together with the user application entitlements. The dispatcher 346 passes the data results obtained from the StarOE 348 to the Web server 344 as shown at 444, which passes the data back to the logon object 342 as shown at 442. The customer 340 is then notified of the logon results as shown as 440.

When the customer 340 is validated properly, the customer is presented with another Web page, referred to as the home page 350, from which the backplane is typically launched. After the user validation, the backplane generally manages the entire user session until the user logs off the "networkMCI Interact." As shown at 448, the backplane initiates a session heartbeat which is used to detect and keep the communications alive between the client platform and the enterprise Intranet site. The backplane also instantiates a COUser object for housekeeping of all client information as received from the StarOE 348. For example, to determine which applications a current customer is entitled to access and to activate only those application options on the home page for enabling the customer to select, the backplane sends a "get application list" message via the Web server 344 and the dispatcher 346 to the StarOE 348 as shown at 448, 444, and 446. The entitlement list for the customer is then sent from the StarOE 348 back to the dispatcher 346, to the Web server 344 and to the backplane at the home page 350 via the path shown at 446, 444, and 448. The application entitlements for the customer are kept in the COUser object for appropriate use by the backplane and for subsequent retrieval by the client applications.

The entitlement information for COUser is stored in a cookie jar 352, maintained in the cookie jar server 32 or the dispatcher server 26 (illustrated in FIGS. 4 and 5). When the Web server receives the entitlement requests from the backplane at the home page 350 or from any other client applications, the Web server 344 makes a connection to the cookie jar 352 and checks if the requested information is included in the cookie jar 352 as shown at 450. The cookie jar 352 is a repository for current customer sessions and the individual session details are included in a cookie including the entitlement information from the OE server 348. During the logon process described above, the OE server 348 may include in its response, the entitlements for the validated customer. The dispatcher 346 transfers the entitlement data to the Web server 344, which translates it into a binary format. The Web server 344 then transmits the binary entitlement data to the cookie jar 352 for storage and retrieval for the duration of a session. Accordingly, if the requested information can be located in the cookie jar 352, no further request to the StarOE 348 may be made. This mechanism cuts down on the response time in processing the request. Although the same information, for example, customer application entitlements or entitlements for corp ids, may be stored in the COUser object and maintained at the client platform as described above, a second check is usually made with the cookie jar 352 via the Web server 344 in order to insure against a corrupted or tampered COUser object's information. Thus, entitlements are typically checked in two places: the client platform 10 via COUser object and the Web server 344 via the cookie jar 352.

Column 24, line 26, to column 25, line 67:

FIG. 13(a) and 13(b) are schematic illustrations showing the message format passed between the dispatcher 26 and the relevant application specific proxy, (FIG. 13(a)) and the message format passed between the application specific proxy back to the Dispatcher 26 (FIG. 13(b)). As shown in FIG. 13(a), all messages between the Dispatcher and the Proxies, in both directions, begin with a common header 150 to allow leverage of common code for processing the messages. A first portion of the header includes the protocol version 165 which may comprise a byte of data for identifying version control for the protocol, i.e., the message format itself, and is intended to prevent undesired mismatches in versions of the dispatcher and proxies. The next portion includes the message length 170 which, preferably, is a 32-bit integer providing the total length of the message including all headers. Next is the echo/ping flag portion 172 that is intended to support a connectivity

test for the dispatcher-proxy connection. For example, when this flag is non-zero, the proxy immediately replies with an echo of the supplied header. There should be no attempt to connect to processes outside the proxy, e.g. the back-end application services. The next portion indicates the Session key 175 which is the unique session key or "cookie" provided by the Web browser and used to uniquely identify the session at the browser. As described above, since the communications middleware is capable of supporting several types of transport mechanisms, the next portion of the common protocol header indicates the message type/mechanism 180 which may be one of four values indicating one of the following four message mechanisms and types: 1) Synchronous transaction, e.g., a binary 0; 2) Asynchronous request, e.g., a binary 1; 3) Asynchronous poll/reply, e.g., a binary 2; 4) bulk transfer, e.g., a binary 3.

Additionally, the common protocol header section includes an indication of dispatcher-assigned serial number 185 that is unique across all dispatcher processes and needs to be coordinated across processes (like the Web cookie (see above)), and, further, is used to allow for failover and process migration and enable multiplexing control between the proxies and dispatcher, if desired. A field 140 indicates the status is unused in the request header but is used in the response header to indicate the success or failure of the requested transaction. More complete error data will be included in the specific error message returned. The status field 140 is included to maintain consistency between requests and replies. As shown in FIG. 13(a), the proxy specific messages 178 are the metadata message requests from the report requester client and can be transmitted via synchronous, asynchronous or bulk transfer mechanisms. Likewise, the proxy specific responses are metadata response messages 180 again, capable of being transmitted via a synchronous, asynchronous or bulk transfer transport mechanism.

It should be understood that the application server proxies can either reside on the dispatcher server 26 itself, or, preferably, can be resident on the middle-tier application servers 40, i.e., the dispatcher front end code can locate proxies resident on other servers.

As mentioned, the proxy validation process includes parsing incoming requests, analyzing them, and confirming that they may include validly formatted messages for the service with acceptable parameters. If necessary, the message is translated into an underlying message or networking protocol. If no errors are found, the proxy then manages the communication with the middle-tier server to actually get the request serviced. The application proxy supports application specific translation and communication

with the back-end application server for both the Web Server (java applet originated) messages and application server messages.

Particularly, in performing the verification, translation and communication functions, the Report Manager server, the Report Scheduler server and Inbox server proxies each employ front end proxy C++ objects and components. For instance, a utils.c program and a C++ components library, is provided for implementing general functions/objects. Various C++ parser objects are invoked which are part of an object class used as a repository for the RM metadata and parses the string it receives. The class has a build member function which reads the string which includes the data to store. After a message is received, the parser object is created in the RMDispatcher.c object which is a file which includes the business logic for handling metadata messages at the back-end. It uses the services of an RMParser class. Upon determining that the client has sent a valid message, the appropriate member function is invoked to service the request. Invocation occurs in MCIRMServerSocket.C when an incoming message is received and is determined not to be a talarian message. RMSErverSocket.c is a class implementing the message management feature in the Report Manager server. Public inheritance is from MCIServerSocket in order to create a specific instance of this object. This object is created in the main loop and is called when a message needs to be sent and received; a Socket.c class implementing client type sockets under Unix using, e.g., TCP/IP or TCP/UDP. Socket.C is inherited by ClientSocket.C:: Socket(theSocketType, thePortNum) and ServerSocket.C:: Socket(theSocketType, thePortNum) when ClientSocket or ServerSocket is created. A ServerSocket.c class implements client type sockets under Unix using either TCP/IP or TCP/UDP. ServerSocket.C is inherited by RMServerSocket when RMServerSocket is created. An InboxParser.c class used as a repository for the RM Metadata. The class' "build"member function reads the string which includes the data to store and the class parses the string it receives. After a message has been received, the MCIInboxParser object is created in inboxutl.c which is a file which includes the functions which process the Inbox requests, i.e, Add, Delete, List, Fetch and Update.

As should be apparent by merely reading the above-copied portions of Devine et al., the portions of Devine et al. that have been applied against independent claim 35 do not disclose the use of "beans" or "a pipe bean", as required by the claim language. In fact, Devine et al. does not even mention the use

of a bean at all. At most, Devine et al. discloses object-oriented classes and objects; however, a bean is not identical nor equivalent with a generic object-oriented class or object because a bean is an object that has particular properties or characteristics. Since Devine et al. does not disclose the use of beans, Devine et al. cannot be used as an anticipatory reference against the claims.

A common rejection was applied against independent claim 35 and dependent claims 4, 16, 27, and 35. Claims 4, 16, and 27 recite that an apparatus, method, or computer program product comprises at least a pipe bean. Since Devine et al. does not disclose the use of beans, Devine et al. cannot be used as an anticipatory reference against claims 4, 16, and 27.

Whereas independent claim 35 is directed to an apparatus that includes a pipe bean, independent claim 44 is directed to an apparatus that includes beans without specifically reciting a pipe bean. However, the rejection of claim 44 pointed to the same portions of Devine et al. as were used against claim 35. Applicant's argument with respect to the rejection of claim 35 is applicable for claim 44.

With respect to the remaining dependent claims, all of the rejections point to the same portions of Devine et al. that were applied against the claims that have been discussed hereinabove. Since Devine et al. does not disclose the use of beans, Devine et al. cannot be used as an anticipatory reference against these claims.

Devine et al. clearly does not disclose features as required by the language of the claims of the present application. As stated at MPEP § 2131: "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The

identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Hence, for this and other reasons, Devine et al. cannot be used as an anticipatory reference, and the rejections of the claims have been overcome, whereby Applicant requests the withdrawal of the rejections.

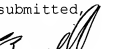
VI. Conclusion

It is respectfully urged that the present patent application is patentable, and Applicant kindly requests a Notice of Allowance.

For any other outstanding matters or issues, the examiner is urged to call or fax the below-listed telephone numbers to expedite the prosecution and examination of this application.

DATE: January 3, 2005

Respectfully submitted,



Joseph R. Burwell
Reg. No. 44,468
ATTORNEY FOR APPLICANT

Law Office of Joseph R. Burwell
P.O. Box 28022
Austin, Texas 78755-8022
Voice: 866-728-3688 (866-PATENT8)
Fax: 866-728-3680 (866-PATENT0)
Email: joe@burwell.biz

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/738,247	12/15/2000	Krishna Kishore Yellepeddy	AUS9-2000-0947 US1	2751

7590 04/21/2005

Law Office of Joseph R. Burwell
P.O. Box 28022
Austin,, TX 78755-8022

EXAMINER	
COLIN, CARL G	
ART UNIT	PAPER NUMBER
2136	

DATE MAILED: 04/21/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/738,247	YELLEPEDDY ET AL.	
	Examiner	Art Unit	
	Carl Colin	2136	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 January 2005* .

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-50* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-50* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 December 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Response to Arguments

1. In response to communications filed on 1/10/2005, applicant amends claims 1, 13, 24, and 33, and claim 8, which was missing has been added. The following claims 1-50 are presented for examination.

2. In response to communications filed on 1/10/2005, the amendment to the specification has been considered and the objection has been withdrawn. Applicant mentions that a set of formal drawings will be mailed separately from the response, which is being faxed. However, no corrected drawings have been received as yet, therefore the objection to the drawings has not been overcome.

3. Applicant's remarks, pages 18-31, filed on 1/10/2005, with respect to the rejection of claims 1-50 have been fully considered but they are moot in view of the new ground(s) of rejection. The amendments to some of the independent claims replacing the reception software to the request implementation software and the addition of new claim 8 have been considered. Applicant has changed the scope of the invention in view of the amended claims, and a new ground of rejection has been made in view of new references as discussed below.

Drawings

4. Figure 3 is objected to as failing to comply with 37 CFR 1.84(p)(5) because it does not include reference numbers (300), (312) and (330) in the description on p. 17, line 18 and 27; page 17, line 25; and page 19, line 19 respectively. Appropriate correction is required.

Figure 3 is objected to as failing to comply with 37 CFR 1.84(p)(5) because it includes the reference sign: 322 not mentioned in the description. Appropriate correction is required.

4.1 Figure 4 is objected to as failing to comply with 37 CFR 1.84(p)(5) because it does not include reference numbers (400) in the description on p. 22, line 12. Appropriate correction is required.

Figure 4 is also objected to as failing to comply with 37 CFR 1.84(p)(5) because it includes the reference sign: 446 not mentioned in the description. Appropriate correction is required.

Applicant is required to carefully review the application to correct such errors.

A proposed drawing correction, corrected drawings, or amendment to the specification to add the reference sign(s) in the description, are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an

international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

5.1 **Claims 35-39 and 41-43** are rejected under 35 U.S.C. 102(e) as being anticipated by US Patent 6,751,735 to **Schell et al**.

5.2 **As per claim 35, Schell et al** discloses an apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising: a plurality of modules communicatively coupled to one another and responsive to events generated that meet the recitation of a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans, for example (see column 9, lines 9-30 and figures 5-6); and discloses a root certifier, a CMC signature root, and other entities and further discloses a CMC signature root that propagates events to another of the plurality of the modules that meets the recitation of at least one of the plurality of beans comprising a pipe bean that propagates an event to another of the plurality of beans, for example (see column 19, lines 1-10 and lines 50-67).

As per claim 36, Schell et al discloses a key generation module as an end module that meets the recitation of the at least one bean comprising a sink bean, the sink bean responsive to propagated events and consuming such propagated events, for example (see column 23, lines 60-65).

As per claim 37, Schell et al discloses the limitation of wherein the pipe bean passes the event to another bean unaltered, for example (see column 26, lines 13-30). **Schell et al** discloses some modules that provide no cryptographic modification that meets the recitation of wherein the pipe bean passes the event to the another bean unaltered.

As per claim 38, Schell et al discloses the limitation of the at least one bean comprising a bean that alters the request, for example (see column 19, lines 51-67). **Schell et al** also discloses one of the pluralities of modules verifying the certificate using a public key. In another embodiment, **Schell et al** discloses a server key generated by the key generation module used for wrapping secret keys used for signing certificates, before being passed to another module (column 23, line 49 through column 24, line 5).

As per claim 39, Schell et al discloses the limitation of further comprising a server bean, the server bean responsive to requests from the distributed processing system (column 11, line 59 through column 12, line 7 and column 14, line 54 through column 15, line 12).

As per claim 41, Schell et al discloses the limitation of further comprising a generation bean, the generation bean generating a digital certificate in response to an event, for example (see column 16, lines 45-56).

As per claim 42, Schell et al discloses the limitation of the at least one bean comprising a bean that publishes information regarding the request, for example (see column 24, lines 8-16).

As per claim 43, Schell et al discloses the limitation of further comprising a filter bean, the filter bean filtering events based upon a predetermined criteria, for example (see column 27, lines 15-20).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6.1 **Claims 1, 3-4, 6-9, 12-13, 15-16, 18-20, 23-24, 26-27, 29-31, 34, 40, 44-50** are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,751,735 to **Schell et al** in view of Non-Patent Literature to **Balfanz et al,** "A Security Infrastructure for Distributed Java Applications"; Security and Privacy, 2000; S&P 2000 Proceedings; 2000 IEEE Symposium on 14-17 May 2000; Pages: 15-26.

6.2 **As per claims 1, 3, 15, and 26, Schell et al** discloses an apparatus for implementing a request regarding a digital certificate in a distributed processing system, the apparatus comprising: any subsequent entity to CMC signature root (see figure 5) that meets the recitation of a request implementation software that implements a response to the request regarding the digital certificate in response to a propagated event object, for example (see column 19, lines 58-67); at least one CMC signature root that meets the recitation of at least one reception bean, communicatively coupled to the request implementation software and the distributed processing system, that generates an event object in response to receiving the request to generate a digital certificate from the distributed processing system, for example (see column 19, lines 51-57). **Schell et al** discloses plurality of modules to generate even object (see figures 4 and 5). **Schell et al** discloses modules to instantiate in real-time but is silent about object-oriented language. **Balfanz et al** in an analogous art discloses an access control system using JAVA permission classes that meets the recitation of a software instantiated in a real time executable object-oriented language (see abstract). The advantage is that it provides a good distribution system and access control mechanism that requests supply credentials that lead to a proof that a request is valid (page 15). Therefore, it would have been obvious to one of ordinary skill in the art at the

time the invention was made modify the apparatus of Schell to implement reception software instantiated in a real time executable object-oriented language such as JAVA permission classes and request-response distributing system as taught by **Balfanz et al**. This modification would have been obvious because one skilled in the art would have been motivated by the suggestions provided by **Balfanz et al** so as to provide a good distribution system and access control mechanism that requests supply credentials that lead to a proof that a request is valid (see page 15).

As per claims 13 and 24, Schell et al discloses a method for implementing a request regarding a digital certificate in a distributed processing system, the method comprising: receiving the request to generate the digital certificate from the distributed processing system in an at least one CMC signature root that meets the recitation of at least one reception bean, for example (see column 19, lines 51-57); the CMC applies a signature that meets the recitation of generating a reception event object in response to step of receiving, for example (see column 19, lines 51-67); forwarding it to another entity that meets the recitation of propagating the reception event object, for example (see column 19, lines 51-67 and column 19, lines 1-10 and 21-29); in another embodiment **Schell et al** discloses third party software for operating selected cryptographic executable for an application associated with a computer (see claim 13); in another embodiment **Schell et al** discloses policy engine that can be linked with CMC to implement any type of filter using rules, attributes, and executables such as key generation, key usage, escrow of keys, etc. (see column 29, lines 20-61) that meets the recitation of selectively implementing a response to the request regarding the digital certificate in response to a propagated event object

in a request implementation software. **Schell et al** modules to instantiate in real-time but is silent about object-oriented language. **Balfanz et al** in an analogous art discloses a security infrastructure for an access control system with PKI using JAVA permission classes that meets the recitation of a software instantiated in a real time executable object-oriented language (see abstract). The advantage is that it provides a good distribution system and access control mechanism that requests supply credentials that lead to a proof that a request is valid (page 15). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made modify the apparatus of Schell to implement reception software instantiated in a real time executable object-oriented language such as JAVA permission classes and request-response distributing system as taught by **Balfanz et al**. This modification would have been obvious because one skilled in the art would have been motivated by the suggestions provided by **Balfanz et al** so as to provide a good distribution system and access control mechanism that requests supply credentials that lead to a proof that a request is valid (see page 15).

As per claim 44, Schell et al discloses an apparatus for implementing a public key infrastructure in a distributed processing system, the apparatus comprising: a plurality of modules communicatively coupled to one another and responsive to events generated that meet the recitation of a plurality of beans, the beans communicatively coupled to one another and responsive to events generated by the plurality of beans, for example (see column 9, lines 9-30 and figures 5-6); **Schell et al** discloses for instance executables within a policy engine that may be used to perform several functions that meets the recitation of respective events generated by the plurality of beans subclassing from a base class event, for example (see column 26, lines 1-

12). Java language is well known in the art as a platform that includes groups of classes and subclasses can be generated from base class event. **Balfanz et al** in an analogous art discloses a security infrastructure for an access control system with PKI using JAVA permission classes as mentioned in claim 1. Therefore, claim 44 is rejected on the same rationale as the rejection of claim 1.

As per claims 4, 16, 27, Schell et al discloses a CMC signature root that meets the recitation of at least one bean comprising a pipe bean, for example (see column 19, lines 1-10 and lines 58-67).

As per claims 6, 18, and 29, Schell et al discloses the limitation of the at least one bean comprising a bean that alters the request, for example (see column 19, lines 51-67). **Schell et al** also discloses one of the pluralities of modules verifying the certificate using a public key. In another embodiment, **Schell et al** discloses a server key generated by the key generation module used for wrapping secret keys used for signing certificates, before being passed to another module (column 23, line 49 through column 24, line 5).

As per claims 7, 19, and 30, Schell et al discloses the limitation of the at least one bean comprising a bean that publishes information regarding the request, for example (see column 24, lines 8-16).

As per claim 8, Schell et al discloses a CMC signature root that meets the recitation of at least one bean comprising a pipe bean, for example (see column 19, lines 1-10 and lines 58-67) and also discloses an end module (152d or 152e) (see column 19, lines 22-23) that meets the recitation of sink bean, for example (see figure 5).

As per claims 9, 20, and 31, Schell et al discloses a key generation module as an end module that meets the recitation of the at least one bean comprising a sink bean, the sink bean responsive to propagated events and consuming such propagated events, for example (see column 23, lines 60-65).

As per claims 12, 23, and 34, Schell et al discloses the limitation of the certificate generation software comprising legacy software, for example (see column 28, lines 49-57).

As per claims 45-46, the combination of **Schell et al** and **Balfanz et al** discloses the limitation of wherein the beans and events are written in a cross platform language, the cross platform language is JAVA, for example (see **Balfanz et al,** abstract). Therefore, they are rejected on the same rationale as the rejection of claim 44 above.

As per claim 47, Schell et al discloses the limitation of the at least one bean comprising a bean that publishes information regarding the request, for example (see column 24, lines 8-16).

As per claim 48, Schell et al discloses the limitation of further comprising a generation bean, the generation bean generating a digital certificate in response to an event, for example (see column 16, lines 45-56).

As per claim 49, Schell et al discloses the limitation of further comprising a server bean, the server bean responsive to requests from the distributed processing system (column 11, line 59 through column 12, line 7 and column 14, line 54 through column 15, line 12).

As per claims 40 and 50, Schell et al discloses the limitation of further comprising a client bean, the client bean responsive to events from the other beans and generating requests to the distributed processing system, for example (see **Balfanz et al,** pages 24-25, section 6).

7. **Claims 2, 5, 10, 11, 14, 17, 21, 22, 25, 28, 32, and 33** are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 6,751,735 to **Schell et al** in view of Non-Patent Literature to **Balfanz et al,** "A Security Infrastructure for Distributed Java Applications"; Security and Privacy, 2000; S&P 2000 Proceedings; 2000 IEEE Symposium on 14-17 May 2000; Pages: 15-26 as applied to claims 1, 13, and 24 and further in view of US Patent Publication US 2001/0001877 to **French et al**.

7.1 **As per claims 2, 14, and 25, Schell et al** discloses plurality of modules and discloses different attributes and functionalities associated with each module (column 26, lines 6-67; see also column 27, line 26 through column 28, line 2). **Schell et al** also discloses that formats may

be governed by policy. **Schell et al** does not explicitly disclose that the events are generated in response to requests of different formats. **French et al** in an analogous art discloses a network authentication system that provides verification of identity and other attributes of a network user to conduct a transaction; a preprocessing stage is employed to ensure correct formatting of the input information. **French et al** discloses generating an event in response to requests of different formats (see page 4, paragraphs 71-75). **French et al** further discloses that one of the advantages of the preprocessing is the ability to process as much requested data as possible from separate data sources and to reduce false negatives due to inconsistencies of mismatched information applied against known data sources (see page 4, paragraphs 72-73). Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made modify the method and apparatus as combined above to have each of the plurality of the modules disclosed in Schell to generate an event in response to requests of differing formats as taught by **French et al**. This modification would have been obvious because one skilled in the art would have been motivated by the suggestions provided by **French et al** so as to provide a consistent data formatting between the information supplied by the user and what is expected from the data sources and the ability to process as much requested data as possible from separate data sources and to reduce false negatives due to inconsistencies of mismatched information applied against known data sources (see page 4, paragraphs 72-73).

As per claims 5, 17, and 28, the combination of **Schell et al, Balfanz et al, and French et al** discloses the limitation of the at least one bean comprising a bean implementing a test on

the request, for example (see **French et al**, page 4, paragraphs 076-077). Therefore, they are rejected on the same rationale as the rejection of claims 2, 14, and 25 above.

As per claims 10, 21, and 32, the combination of **Schell et al, Balfanz et al, and French et al** discloses the limitation of the at least one bean comprising a client bean that propagates a request in a first format, for example (see **French et al**, pages 4, paragraphs 071-073). Therefore, they are rejected on the same rationale as the rejection of claims 2, 14, and 25 above.

As per claims 11, 22, and 33, the combination of **Schell et al, Balfanz et al, and French et al** discloses the limitation of the at least one bean comprising another client bean that propagates a request in a second format another client bean that propagates a request in a second format, for example (see **French et al**, pages 4, paragraphs 071-073). Therefore, they are rejected on the same rationale as the rejection of claims 2, 14, and 25 above.

Conclusion

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

8.1 Any inquiry concerning this communication or earlier communications from the examiner should be directed to Carl Colin whose telephone number is 571-272-3862. The examiner can normally be reached on Monday through Thursday, 8:00-6:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ayaz Sheikh can be reached on 571-272-3795. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Cc

Carl Colin
Patent Examiner
April 11, 2005

AYAZ SHEIKH
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

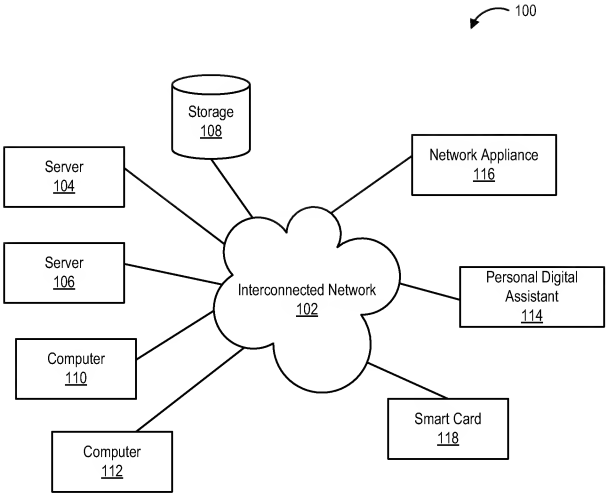

100
Storage
108
Server
104
Server
106
Computer
110
Computer
112
Network Appliance
116
Interconnected Network
102
Personal Digital
Assistant
114
Smart Card
118

FIGURE 1

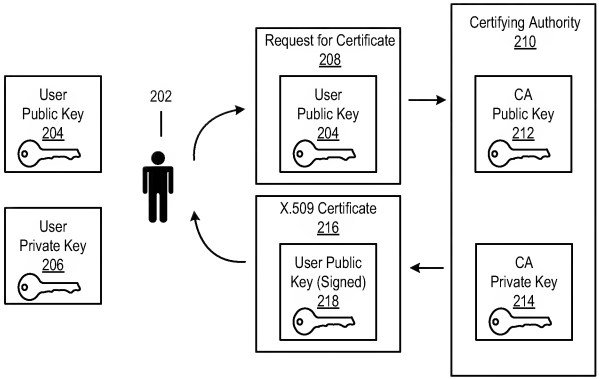

User
Public Key
204
User
Private Key
206
202
Request for Certificate
208
User
Public Key
204
X.509 Certificate
216
User Public
Key (Signed)
218
Certifying Authority
210
CA
Public Key
212
CA
Private Key
214

FIGURE 2

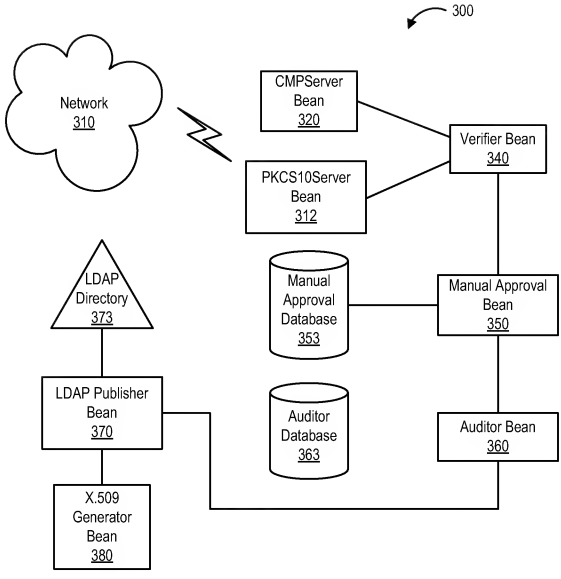

300
Network
310
CMPServer
Bean
320
PKCS10Server
Bean
312
Verifier Bean
340
LDAP
Directory
373
Manual
Approval
Database
353
Manual Approval
Bean
350
LDAP Publisher
Bean
370
Auditor
Database
363
Auditor Bean
360
X.509
Generator
Bean
380

FIGURE 3

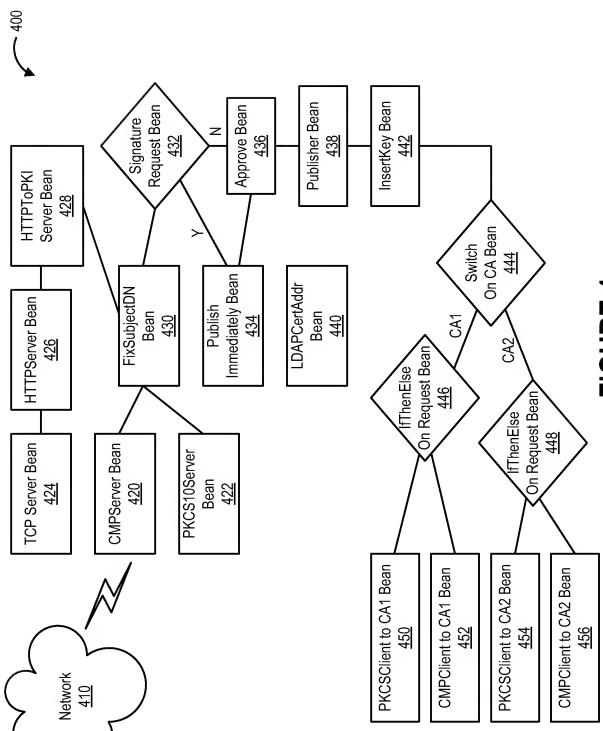

400
Network
410
TCP Server Bean
424
HTTPServer Bean
426
HTTPToPKI
Server Bean
428
CMPServer Bean
420
PKCS10Server
Bean
422
FixSubjectDN
Bean
430
Signature
Request Bean
432
Y
N
Publish
Immediately Bean
434
Approve Bean
436
LDAPCertAddr
Bean
440
Publisher Bean
438
InsertKey Bean
442
Switch
On CA Bean
444
CA1
CA2
IfThenElse
On Request Bean
446
IfThenElse
On Request Bean
448
PKCSClient to CA1 Bean
450
CMPClient to CA1 Bean
452
PKCSClient to CA2 Bean
454
CMPClient to CA2 Bean
456

FIGURE 4

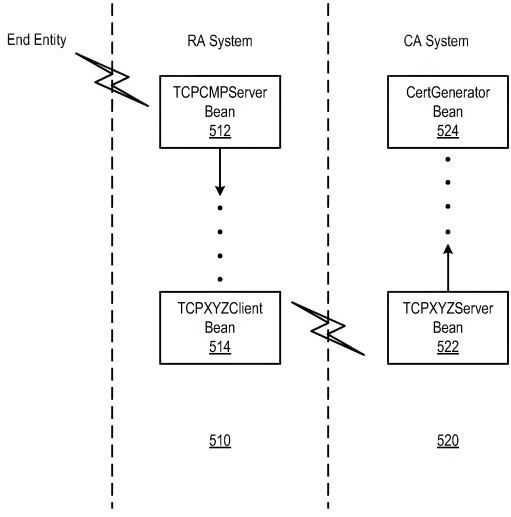

End Entity
RA System
CA System
TCPCMPServer Bean 512
CertGenerator Bean 524
TCPXYZClient Bean 514
TCPXYZServer Bean 522
510
520

FIGURE 5

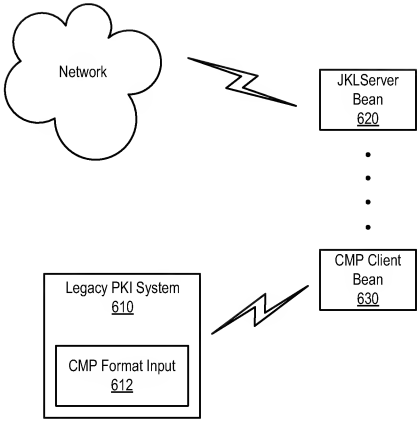

Network
JKLServer Bean 620
CMP Client Bean 630
Legacy PKI System 610
CMP Format Input 612

FIGURE 6

- PkEvent
  - PkReqEvent
    - PkCertReqEvent
      - PkInitialReqEvent
      - PkSecondaryReqEvent
      - PkUpdateReqEvent
      - PkCrossCertReqEvent
    - PkRevocationReqEvent
    - PkHttpReqEvent
    - PkKeyRecoveryReqEvent
  - PkRepEvent
    - PkCertRepEvent
      - PkInitialRepEvent
      - PkSecondaryRepEvent
      - PkUpdateRepEvent
      - PkCrossCertRepEvent
    - PkRevocationReqEvent
    - PkHttpRepEvent
    - PkKeyRecoveryRepEvent

FIGURE 7

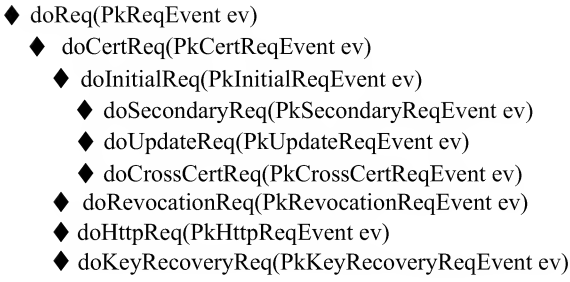

doReq(PkReqEvent ev)
doCertReq(PkCertReqEvent ev)
doInitialReq(PkInitialReqEvent ev)
doSecondaryReq(PkSecondaryReqEvent ev)
doUpdateReq(PkUpdateReqEvent ev)
doCrossCertReq(PkCrossCertReqEvent ev)
doRevocationReq(PkRevocationReqEvent ev)
doHttpReq(PkHttpReqEvent ev)
doKeyRecoveryReq(PkKeyRecoveryReqEvent ev)

FIGURE 8

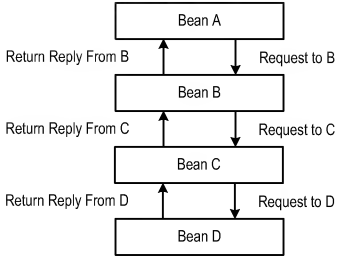

Bean A
Return Reply From B
Request to B
Bean B
Return Reply From C
Request to C
Bean C
Return Reply From D
Request to D
Bean D

FIGURE 9